

# LTPW1

## Capítulo 06

# Fazendo umas contas com JS

Você é daquele time de pessoas que odeia Matemática, ou é dos meus e adora essa Ciência maravilhosa? Independente do lado que você esteja, com uma coisa temos que concordar: fazer cálculos simples faz parte do nosso dia-a-dia! Coisas primordiais como calcular uma média, fazer a conta do troco da padaria ou quantos anos ainda faltam pra você se aposentar (não faça essa com tanta frequência, senão vai acabar ficando triste). Nesse capítulo, vamos ver como podemos fazer conta em nossos scripts.



Você tem todo o direito de usar esse material para seu próprio aprendizado. Professores também podem ter acesso a todo o conteúdo e usá-los com seus alunos. Porém todos o que usarem esse material - seja para qual for a finalidade - deverão manter a referência ao material original, criado pelo **Prof. Gustavo Guanabara** e disponível no endereço do seu repositório público <https://github.com/gustavoguanabara/>. Este material não poderá ser utilizado em nenhuma hipótese para ser replicada - integral ou parcialmente - por autores/editoras para criar livros ou apostilas, com finalidade de obter ganho financeiro com ele.



# A Matemática é como o cachorro do meu vizinho

Calma! Não estou ofendendo o meu vizinho chamando de cachorro! O que estou querendo explicar é que meu vizinho tem um cachorro daqueles bem pequenos e que acha que é um cão gigante. Já viu um desses?



Quando eu vou sair de casa para ir dar aula e o cachorro me vê, começa a latir raivosamente e mostrar os dentes pra mim. É como se ele estivesse querendo me dizer: *"chega mais perto, e eu arranco o seu braço na dentada"*.

Acontece que eu sei que ele não pode me fazer mal. Aí falo o nome dele, me aproximo, abaixo e coloco a mão na sua cabeça. Ele imediatamente começa a abanar o rabo e às vezes até deita no chão de tanta alegria.

Matemática também é assim. Ela assusta às vezes. Se você se intimidar ela fica assustadora. Mas se você enfrentar o desafio, ela vai ser dócil contigo! Pode acreditar em mim!

## Operador de Atribuição (parte 1)

A coisa mais simples que você pode fazer em JS é a atribuição. Atribuir significa jogar um valor ou o resultado de uma expressão para dentro de uma variável. Vamos a um exemplo bem simples:

```
1 let n1 = 4
2 let n2 = 8
3 let n3 = n1 + n2
```

As três linhas ao lado possuem o mesmo operador de atribuição, que é o símbolo de `=`. Vamos fazer um acordo aqui: sempre que você encontrar um símbolo desses no código, vai ler como se fosse a palavra **"recebe"**.

Sendo assim, vamos ler as linhas assim:

- 1 - "a variável `n1` **recebe** o valor 4"
- 2 - "a variável `n2` **recebe** o valor 8"
- 3 - "a variável `n3` **recebe** o resultado da expressão `n1+n2` (o valor 12)"



**E O SINAL DE IGUAL?** Você pode estar se perguntando: se apenas um sinal `=` significa "recebe", como vamos representar o operador de "igual"? A resposta é simples: Em JavaScript, o "igual" é representado como `==` (dois sinais na sequência, sem espaço entre eles) ou como `===` (três sinais consecutivos, ou igualdade estrita).

Ainda existem outros operadores de atribuição, mas veremos isso mais tarde na parte 2 desse assunto, combinado?

# Operadores Aritméticos

Os operadores desse tipo servem para fazer as contas básicas que estamos acostumados no nosso dia-a-dia:

| Operador | Função            | Exemplo | Resultado |
|----------|-------------------|---------|-----------|
| ( )      | Agrupamento       | (5+3)/2 | 4         |
| +        | Positivo unário   | +5      | 5         |
| -        | Negativo unário   | -4      | -4        |
| **       | Exponenciação     | 5 ** 2  | 25        |
| *        | Multiplicação     | 3 * 2   | 6         |
| /        | Divisão           | 10 / 2  | 5         |
| %        | Resto da divisão  | 5 % 2   | 1         |
| +        | Adição Binária    | 5 + 3   | 8         |
| -        | Subtração Binária | 8 - 5   | 3         |

O primeiro operador (Agrupamento) não é exatamente um "operador aritmético", mas adicionei aqui na lista para efeitos práticos. Usamos muito eles para forçar a execução de uma operação antes de outra. No exemplo que demos, se resolvermos  $5+3/2$  o resultado será 6.5, já que em uma expressão aritmética com soma binária e divisão, resolvemos primeiro a divisão. Chamamos isso de **ordem de precedência**.

Como você já deve ter percebido, essa tal de "ordem de precedência" indica quais operadores serão executados primeiro. Nem tudo é resolvido na ordem em que aparece na expressão. Os que estão acima são mais poderosos. Nos lugares onde aparecerem o símbolo [ os operadores possuem a mesma ordem e são poderosos da mesma maneira, sendo assim devem ser resolvidos da esquerda para a direita. Sempre que você se deparar com uma expressão aritmética, consulte a tabela acima. Se preferir, pode imprimir essa página, recortar a tabela e colar em um lugar de destaque.

Vejamos um exemplo, comparando a expressão  $5+3/2$  com a expressão  $(5+3)/2$ :

$$5 + 3 / 2$$

$$(5 + 3) / 2$$

No primeiro caso, quando não coloquei parênteses, a ordem de operação será para aquele operador onde a precedência vem antes, no caso a divisão, que resolveremos primeiro. Dessa maneira, o resultado da expressão à esquerda seria 6.5

Já na expressão da direita existem os parênteses - que possuem a precedência mais alta - e começaremos por eles, realizando a soma antes e depois a divisão. O resultado da segunda expressão, seria então 4.



**DICA:** Analise os operadores em uma expressão, como se fosse uma fila de banco. Não importa muito a ordem visual da fila, os idosos serão atendidos primeiro. Eles têm preferência! Na tabela, quem tem mais preferência são os operadores que estão mais pra cima. Os Operadores que estão ligados pelo sinal de [ possuem a mesma preferência, assim quem será atendido primeiro é quem estiver à esquerda.

## Não entendi os operadores % e \* \*

Normalmente meus alunos não entendem de cara o significado do operador %. "-Mas isso não é pra calcular porcentagem?", eles perguntam.

O símbolo % não tem essa finalidade, assim como o x não calcula a multiplicação. Programação tem suas particularidades, já vai se acostumando!

Vamos começar com um cálculo simples:  $134 \% 5$ . Armei a conta aqui embaixo e vamos resolvê-la até encontrar o valor inteiro da divisão (sem usar a vírgula).

$$\begin{array}{r} 134' \\ 34 \overline{) 26} \\ 4 \end{array}$$

Quando chegamos ao resto 4, só conseguimos continuar a divisão colocando um zero a mais ao lado do resto e uma vírgula no final do 26, não é? Não queremos isso, pois estamos buscando pela **divisão inteira**.

Sendo assim,  $134 \% 5 = 4$ . Sacou?

Agora vamos nos focar no operador \*\*, que mais parece que estou multiplicando duas vezes. Mas não é bem assim. Esse operador na verdade calcula o valor de uma exponenciação ou **potência**.

Por exemplo, se quisermos calcular  $12^2$ , basta escrever  $12 ** 2$  e o motor JavaScript vai entender.

Agora que você já entendeu o funcionamento de cada operador aritmético do JavaScript, vamos colocar um pouco a mão na massa para treinar a sua aplicação em expressões mais complexas.

# Vamos praticar um pouquinho?

Vamos ver se você entendeu o que é ordem de precedência. Fique sempre de olho na tabela da página anterior e marque quem vem primeiro em cada expressão antes de resolvê-la de fato. Eu facilitei e já fiz o primeiro pra você. As próximas são contigo!

$$8 * 3 + 2 / 2 = 25$$

$$1 + 3 ** 2 / 9 = 2$$

$$1 + 3 ** 2 / 9 = 2$$

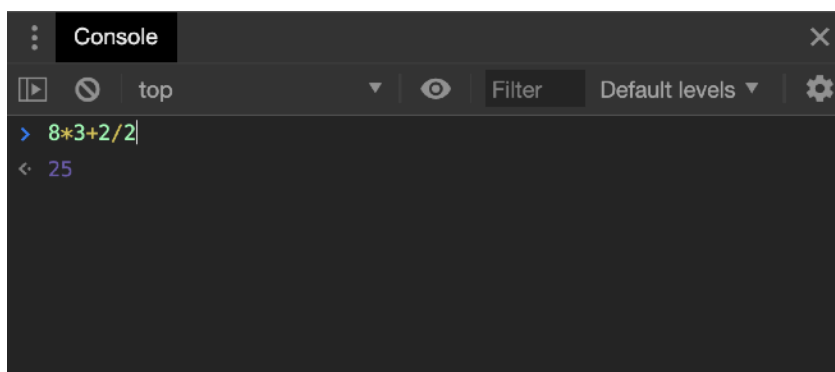
$$-3 + 4 * ( 10 \% 4 ) = 5$$

$$( 8 - 2 * 3 ) \% 1 * 2 = 0$$

## Quer saber se suas contas estão certas?

Você se lembra quantas vezes eu te disse que o **DevTools** do Google Chrome é seu melhor amigo? Pois ele vai te ajudar de novo!

Abra no seu Google Chrome as **Ferramentas do Desenvolvedor** do jeitinho que eu te ensinei no **Capítulo 03**, na seção "*Não funcionou? Não se desespere!*", ou simplesmente aperte Ctrl+Alt+I. Vá até a parte do **Console** e digite qualquer fórmula diretamente.



Viu? Confia no **DevTools**! Ele está sempre ao seu lado! No momento em que você achar que as coisas pararam de funcionar ou sentir que algo deu errado, é só abrir essa janelinha mágica e ela quase sempre te dará uma resposta, ou pelo menos uma dica que te levará ao resultado.

# Operadores de Atribuição (parte 2)

Agora que você já conhece os operadores aritméticos básicos, podemos combiná-los aos operadores de atribuição para realizar tarefas que são muito comuns. Por exemplo, uma expressão que faremos com muita frequência é a operação de acumulador. Algo como `soma = soma + num`. Podemos simplificá-la usando operadores de simplificação aritmética:

| Operador         | Exemplo              | Equivale a...           |
|------------------|----------------------|-------------------------|
| <code>+=</code>  | <code>a += b</code>  | <code>a = a + b</code>  |
| <code>-=</code>  | <code>a -= b</code>  | <code>a = a - b</code>  |
| <code>*=</code>  | <code>a *= b</code>  | <code>a = a * b</code>  |
| <code>/=</code>  | <code>a /= b</code>  | <code>a = a / b</code>  |
| <code>**=</code> | <code>a **= b</code> | <code>a = a ** b</code> |
| <code>%=</code>  | <code>a %= b</code>  | <code>a = a % b</code>  |

Pode ser que você não tenha entendido muito bem esses operadores agora. Não se preocupe! Mais pra frente voltamos neles e tudo ficará mais claro.

## Operadores de incremento/decremento

Ainda nos mantendo no mesmo assunto de simplificar expressões muito comuns usadas no dia-a-dia do programador. Outro tipo de expressão bastante usada é o incremento (somar uma unidade a uma variável) ou decremento (diminuir uma unidade a uma variável).

| Operador        | Exemplo           | Equivale a...          | Poderia usar        |
|-----------------|-------------------|------------------------|---------------------|
| <code>++</code> | <code>x ++</code> | <code>x = x + 1</code> | <code>x += 1</code> |
| <code>--</code> | <code>x --</code> | <code>x = x - 1</code> | <code>x -= 1</code> |

## E se eu precisar fazer umas contas mais complexas?

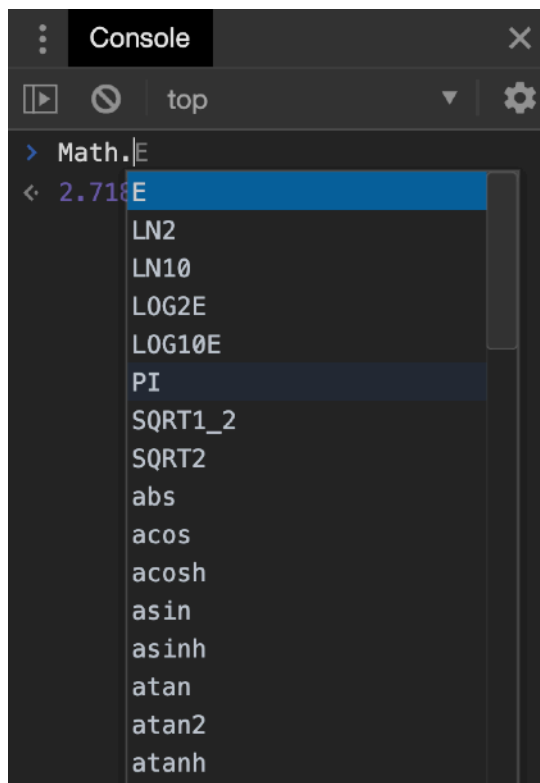
Com o uso dos operadores aritméticos e um bom conhecimento de Matemática já é possível resolver quase tudo. Porém, nem sempre queremos ficar estudando teoremas e correndo atrás de resolver tudo manualmente.

Pensando nisso, o JavaScript tem um objeto global especial chamado **Math** que possui vários métodos e atributos internos para operações Matemáticas (como o próprio nome já sugere).

Aproveita que o seu **DevTools** já deve estar aberto, vai até a janela de **Console** e digita **Math**. (não se esqueça de colocar a letra M maiúscula e o ponto logo depois, sem espaço nenhum).

Olha só ao lado a quantidade de métodos e atributos dentro do objeto **Math** e que podem ser utilizados em seus códigos a partir de agora.

Vou colocar aqui algumas coisas que podem ser feitas usando **Math**, usando alguns exemplos de comandos que você pode digitar no Console também, se quiser ver o resultado (só não se esqueça que o JS diferencia maiúsculas e minúsculas):



| Digite isso...             | ... o resultado será                                |
|----------------------------|---|
| Math.PI                    | O valor de PI com precisão de 15 dígitos            |
| Math.abs(-200)             | O módulo ou valor absoluto do número                |
| Math.sqrt(81)              | A raiz quadrada do número informado                 |
| Math.cbrt(27)              | A raiz cúbica do número informado                   |
| Math.ceil(12.2)            | Arredonda pra cima o número informado               |
| Math.floor(15.7)           | Arredonda pra baixo o número                        |
| Math.round(18.5)           | Faz o arredondamento aritmético tradicional         |
| Math.trunc(22.9234)        | Trunca o valor, eliminando toda a parte fracionária |
| Math.min(5, 9, 2, 6, 1, 4) | Indica qual foi o menor valor informado             |
| Math.max(5, 9, 2, 6, 1, 4) | Indica qual foi o maior valor informado             |
| Math.random()              | Gera um número aleatório entre 0 e 1                |



**REFERÊNCIA COMPLETA:** Para uma lista completa, consulte o manual oficial da **Fundação Mozilla** no endereço abaixo:

[https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global\\_Objects/Math](https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/Math)

# E o cálculo de porcentagem?

Lá na capa desse capítulo eu perguntei se você é do tipo de pessoa que odeia ou adora Matemática. Se você foi do segundo grupo, não precisa se preocupar com essa sessão. Agora, se você é daqueles que não curte a ciência dos números, provavelmente fica travado(a) quando o assunto é porcentagem.



**RESUMINDO E SIMPLIFICANDO:** De forma fácil, uma porcentagem é uma **razão com denominador 100** (por isso se chama "por-100-to"). Ele calcula uma medida proporcional de um valor em relação ao "todo" ou 100%.

A maneira mais simples é resolver usando uma **regra de três** simples. Vamos ver dois exemplos:

**EXEMPLO 01:** Uma loja de tecnologia vende um computador por **R\$1850,00**. Na época da Black Friday, o dono da loja autorizou um desconto de 20% no preço do PC para aumentar as vendas e liberar estoque. Qual será o valor do PC na Black Friday?

**EXEMPLO 02:** José é um ótimo funcionário. Chega sempre na hora certa e quase não falta ao trabalho. Seu salário é de **R\$3480,00** há 2 anos. No próximo mês, o chefe do José resolveu dar um aumento a ele e disse que seu salário vai subir 13%. Qual será o valor do novo salário do José?

Nos dois casos, são citados valores percentuais (20% de desconto no primeiro caso e 13% de aumento no segundo). Vamos usar como base o primeiro exemplo:

O pensamento é o seguinte: o PC custa R\$1850,00, e isso corresponde a 100% do preço. Quanto será que vai custar (x) 20% desse valor?

$$\begin{array}{lcl} \text{R\$ } 1850 & \xrightarrow{\quad\quad\quad} & 100\% \\ \text{R\$ } x & \xrightarrow{\quad\quad\quad} & 20\% \end{array}$$

Vamos resolver essa regra de três multiplicando apenas o valor, de forma cruzada:

$$x \cdot 100 = 1850 \cdot 20$$

Agora vamos deixar apenas x em evidência do lado esquerdo da expressão, passando o 100 para o lado direito, que deixa de multiplicar e passa a dividir:

$$x = \frac{1850 \cdot 20}{100}$$

Então a fórmula acima calcula quanto é 20% de R\$1850,00. Aí está a nossa **razão com denominador 100** que estávamos procurando.

Agora podemos calcular o novo preço do produto, baseado nos 20% calculados:



$$\text{novo} = 1850 - x$$

De forma similar, podemos calcular os 13% do salário do José:

$$x = \frac{3480 \cdot 13}{100}$$

Viu como é simples? Agora podemos calcular o novo salário:

$$\text{novo} = 3480 + x$$

Note que dependendo do tipo de reajuste, usamos o operador mais coerente para cada caso. No exemplo 1, tínhamos um DESCONTO, logo subtraímos o valor de x. Já no exemplo 2, tínhamos um AUMENTO, e por isso somamos o valor de x.

Se quisermos transformar isso tudo em um código simples em JavaScript, podemos escrever:

#### EXEMPLO 01 em JavaScript

```
1 let preço = 1850
2 let porc = 20
3 let valor = (preço * porc) / 100
4 let novo = preço - valor
```

#### EXEMPLO 02 em JavaScript

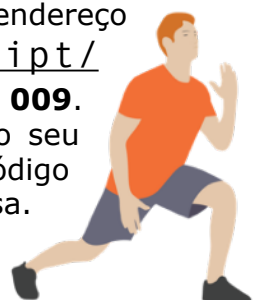
```
1 let sal = 3480
2 let porc = 13
3 let valor = (sal * porc) / 100
4 let novo = sal + valor
```



**NÃO ESQUEÇA:** Durante a programação, o símbolo de % usado no código serve para calcular o resto da divisão inteira, não tem nada a ver com a porcentagem.

## Muitos exercícios

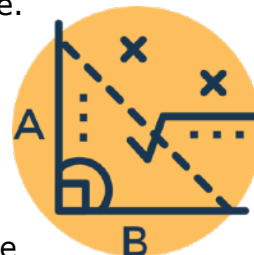
Agora chegou a hora de praticar MUITO. Acesse agora mesmo o endereço <https://gustavoguanabara.github.io/javascript/exercicios/> e execute os **exercícios 005, 006, 007, 008 e 009**. Depois crie as pastas **ex005, ex006, ex007, ex008 e ex009** no seu computador e tente fazer esses mesmos exercícios sem copiar o código que eu criei. Nesse momento, a prática é algo que você mais precisa. Se por acaso ficar difícil, pode acessar o repositório público de HTML e CSS e dar uma olhada nos comandos, mas **EVITE COPIAR**.



# Por enquanto é isso

Espero que esse capítulo tenha deixado tudo bem claro pra você. Guarde essa parte do seu curso com carinho e se preferir, pode até imprimir o capítulo inteiro para quando for precisar fazer uma consulta rápida.

Matemática **não é** um bicho de sete cabeças. Treine bastante no **DevTools** do seu Google Chrome e experimente criar expressões que você está acostumado(a) no seu dia-a-dia. Não deixe também de fazer os exercícios do número 005 até o 009.



## Tenho uns desafios pra você!

Lá no repositório, além do material em PDF e dos códigos dos exercícios 100% disponíveis, também disponibilizamos alguns **desafios** que devem ser resolvidos. Esses desafios não incluem o código original e você deve tentar chegar à resposta sem copiar nenhum código.



Com todo o conteúdo que vimos até essa aula, você já pode resolver os seguintes desafios:

- **d003:** Antecessor e sucessor
- **d004:** Calculando o troco
- **d005:** Conversor de medidas
- **d006:** Conversor de temperatura
- **d007:** Conversor de moedas
- **d008:** Calculando desconto de 10%
- **d009:** Aumentando o salário
- **d010:** Resolvendo Bhaskara

Acesse o repositório público, abra a área do curso de JavaScript e clique no link de acesso aos desafios. Manda ver! Só não fica pedindo a resposta! Você consegue resolver isso sozinho(a)!

Repositório em: <https://gustavoguanabara.github.io>

## Eu já falei sobre isso no YouTube?

Eu sei que às vezes as pessoas gostam mais de assistir vídeos do que ler livros, e é por isso que eu lanço há anos materiais no canal Curso em Vídeo no YouTube. O link que vou compartilhar contigo vai diretamente para a playlist completa do curso que já está totalmente disponível e mostra todos os procedimentos passo-a-passo. Acesse agora mesmo!



Curso em Vídeo: [https://www.youtube.com/playlist?list=PLHz\\_AreHm4dlsK3Nr9GVvXCbpQyHQ11o1](https://www.youtube.com/playlist?list=PLHz_AreHm4dlsK3Nr9GVvXCbpQyHQ11o1)